Research Report 218

A Probabilistic Duration Calculus*

Zhiming Liu, Anders P Ravn,
Erling V Sørensen and Chaochen Zhou

RR218

This paper presents a calculus that enables a designer of an embedded, real-time system to reason about and calculate whether a given requirement will hold with a sufficient high probability for given failure probabilities for components used in the design of the system. The main idea is to specify requirements and design in Duration Calculus, a real-time, interval logic, to define satisfaction probabilities for formulas in this calculus, and establish a calculus with rules that support calculation of the probability for a composite formula from probabilities of its constituents. This ensures that reasoning about probabilities is consistent with requirements and design decisions. We thus avoid introducing separate models for requirements and reliability analysis. The system model is a finite automaton with fixed transition probabilities. This defines discrete Markov processes as basis for the calulus.

Keywords: duration calculus, real-time systems, probabilistic automata, satisfaction probability.

Department of Computer Science
University of Warwick
Coventry CV4 7AL
United Kingdom

May 1992

A Probabilistic Duration Calculus*

Zhiming Liu[†‡] Anders P. Ravn[†] Erling V. Sørensen[†] and Chaochen Zhou[†§¶]

Abstract: This paper presents a calculus that enables a designer of an embedded, real-time system to reason about and calculate whether a given requirement will hold with a sufficient high probability for given failure probabilities for components used in the design of the system. The main idea is to specify requirements and design in Duration Calculus, a real-time, interval logic, to define satisfaction probabilities for formulas in this calculus, and establish a calculus with rules that support calculation of the probability for a composite formula from probabilities of its constituents. This ensures that reasoning about probabilities is consistent with requirements and design decisions. We thus avoid introducing separate models for requirements and reliability analysis. The system model is a finite automaton with fixed transition probabilities. This defines discrete Markov processes as basis for the calculus.

Keywords: duration calculus, real-time systems, probabilistic automata, satisfaction probability.

1 Introduction

Requirements for an embedded, real-time system include functional and safety properties. Consider for instance an on-off gas burner [SNH91]. It is required to turn the flame on or off a short time after requested to do so by a thermostat. It must also prevent excessive leak of gas to the environment. The latter requirement can be stated as an integrated constraint: the duration of leaking states should only be a small proportion of any 1 minute interval.

Such a system can be modelled by a dynamic system where a state changes over time. In the gas burner example, we could for instance introduce a state *Leak*, that varies with time. A design for discrete control of the system will then be given by constraints on the transitions between states. A designer may now use various mathematical techniques to verify that the design satisfies the requirements. Among them the duration calculus [ZHR92] is recently found promising for reasoning about requirements and designs of real-time, embedded systems [HRR91, RR91, SRRZ92]. A summary of this calculus is given in Section 2.

However, a customer or a certification agency may legitimately ask about the dependability of the system in terms of a failure probability within a certain period of time. Such a question cannot be answered from the design or its mathematical model. In order to answer the questions, the designer may choose to develop alternative models, cf. the two tiered approaches used in

*This work was supported in part by **ProCoS** ESPRIT BRA 3104, and by the Danish Technical Research Council under project **RapID**.

[†]Department of Computer Science, Technical University of Denmark, DK-2800, Lyngby, Denmark.

[‡]On leave from Department of Computer Science, University of Warwick, Coventry, CV4 7AL, England.

[§]Also visiting Programming Research Group, Oxford University, England.

[¶]On leave from Software Insititue, Academia Sinica, Beijing, China.

the SIFT project [MSS82], or the stochastic model developed from a state machine model for a design in [SNH91].

A two model approach adds complexity to the design activity because the two have somehow to be updated consistently whenever the design changes. Several researchers have seen that there is a potential for making the design activity simpler by using a unified model in the form of a probabilistic automaton with Markov properties [HJ89, LS89]. In [LS89] an untimed logic for specification is extended by adding probabilities to the combinators; this allows reasoning about untimed probabilistic systems. Time and probabilities are introduced together in [HJ89] which extends the computation tree logic (CTL) of [CES83]. There is however not a proof system for the extended logic, and the expressiveness is somewhat restricted. We have thus found it worthwhile to investigate the development of a probabilistic duration calculus.

Based on probabilistic automata, this paper defines the satisfaction probabilities of duration formulas in the duration calculus, and establishes a corresponding probabilistic calculus. The calculus has a set of axioms and rules that support direct reasoning about and calculation of satisfaction probabilities for formulas specifying a given design.

The probabilistic duration calculus is based on three key ideas. The first is to simulate imperfect systems with probabilistic automata. This is presented in Section 3. The second one, in Section 4, is to extend the model of the duration calculus and define the satisfaction probability of a duration formula by a probabilistic automaton. And the third one is to establish a calculus to calculate and reason about satisfaction probabilities. This is done in section 5. Running examples are given in each section and Section 6 contains a number of examples that illustrate the possible application of the calculus. The conclusion in Section 7 compares this work with related work.

2 The Duration Calculus

This section outlines the duration calculus and its application to specifying real-time systems.

2.1 Time

The original duration calculus [ZHR92, HZ92] uses continuous time. In order to have a simple, well understood probabilistic model (see Section 3), we here assume discrete time. Time is represented by the set N of non-negative integers. A time point is denoted t, t_1, etc. and a *time interval* $[t_1, t_2]$, $t_1 \leq t_2$, represents the set of time points from t_1 to t_2.

2.2 States

We assume a finite non-empty set A of *primitive states*. States, ranged over by P, Q, P_1, Q_1, etc., consist of expressions formed by the following rules:

- Each primitive state $P \in A$ is a state.
- If P and Q are states, then so are $\neg P$, $(P \wedge Q)$, $(P \vee Q)$, $(P \Rightarrow Q)$, $(P \Leftrightarrow Q)$.

A primitive state P is interpreted as a function $I(P) : N \to \{0,1\}$. $I(P)(t) = 1$ means that state P is present at time point t, and $I(P)(t) = 0$ means that state P is not present at time

point t. We assume that when a state is present at time t, it will persist for the next time unit. A *composite state* is interpreted as a function which is defined by the interpretations for the primitive states and the boolean operators.

2.3 Duration

For an arbitrary state P, its *duration* is denoted $\int P$. Given an interpretation I of the states, a duration $\int P$ will be interpreted over time intervals. It denotes the accumulated time P is present within the time interval. So for an arbitrary interval $[t_1, t_2]$, the interpretation $I(\int P)([t_1, t_2])$ is defined as the non-negative integer

$$I(\int P)([t_1, t_2]) = \sum_{t=t_1}^{t_2-1} I(P)(t)$$

where $I(\int P)([t, t]) = 0$. So $\int 1$ always denotes the length of an interval.

The set of *primitive duration terms* consists of variables over the integers Z and durations of states. A *duration term* is either a primitive term or an expression formed from terms by using the usual operators on integers, such as addition $+$ and multiplication $*$.

2.4 Duration Formulas

A *primitive duration formula* is an expression formed from terms by using the usual relational operators on the integers, such as equality $=$ and inequality $<$. A *duration formula* is either a primitive formula or an expression formed from formulas by using the logical operators $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, and the *chop* ; and quantifiers $\forall$, $\exists$ applied to variables ranging over Z.

A duration formula D is satisfied by an interpretation I with an interval $[t_1, t_2]$ just when it is evaluated to *true* [HZ92]. There it is written

$$I, [t_1, t_2] \models D$$

A chopped formula D_1; D_2 is true for I with $[t_1, t_2]$ if there exists a t such that $t_1 \leq t \leq t_2$ and D_1 and D_2 are true respectively with $[t_1, t]$ and $[t, t_2]$ for I.

We define shorthands for some duration formulas which are often used.

Definition 1 *For an arbitrary state P,*

$$\lceil P \rceil \triangleq (\int P = \int 1) \wedge (\int 1 > 0)$$

This means that P holds everywhere in a non-point interval. We use $\lceil\ \rceil$ to denote the predicate which is true only for a point interval.

Definition 2 $\lceil\ \rceil \triangleq \int 1 = 0$

Definition 3 *For a duration formula D,*

$$\Diamond D \triangleq \mathit{true};\ D;\ \mathit{true}$$

This is true of an interval in which D holds for some subinterval of it.

Definition 4 *For a duration formula D,*

$$\Box D \triangleq \neg\Diamond\neg D$$

This is true of an interval where D holds for all subintervals of it.

2.5 Real Time Specifications

The duration calculus has been used to specify real-time constraints of embedded systems [HRR91, RR91, SRRZ92]. In [ZHR92], one of the time critical requirements of a Gas Burner is specified by a formula of the duration calculus denoted as **Req-1**,

$$\textbf{Req-1} \quad \int 1 > 60sec \Rightarrow (20 * \int Leak \leq \int 1)$$

This says that if the interval over which the system is observed is at least one minute, the proportion of time spent in the leak state is no more than one twentieth of the elapsed time.

The requirement is refined into two design decisions

$$\textbf{Des-1} \quad \Box(\lceil Leak \rceil \Rightarrow \int 1 \leq 1sec)$$

$$\textbf{Des-2} \quad \Box(\lceil Leak \rceil;\ \lceil \neg Leak \rceil;\ \lceil Leak \rceil \Rightarrow \int 1 > 32sec)$$

Des-1 says that any leak state must be detected and stopped within one second, and **Des-2** says leak must be seperated by at least 30 seconds.

The correctness of the design is reasoned about by proving the implication

$$\textbf{Des-1} \wedge \textbf{Des-2} \Rightarrow \textbf{Req-1}$$

in the duration calculus [ZHR92].

However, we cannot expect, in practice, a *real* implementation to satisfy the decisions at all time. A real implementation can only satisfy the design decisions with some probability within a given service period. This raises the following problems which are the concerns of this paper. How can we model a real (imperfect) implementation? How can we define and reason about the satisfication probability of a duration formula (requirement or decision)?

3 Imperfect Systems and Probabilistic Automata

We will use a *finite probabilistic automaton* as a mathematical model of the behaviour of an imperfect system in a discrete time domain. Such an automaton is well described by its transition graph. We will continue with the Gas Burner example.

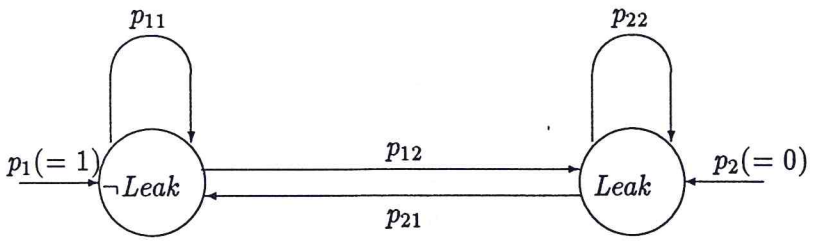

Figure 1: A Gas Burner With Unreliable Detector

3.1 Unreliable Flame Detector

A model of a Gas Burner with an unreliable flame detector can be defined by the transition graph shown in Figure 1. In this Gas Burner, *Leak* is the only primitive state considered, and $\neg$*Leak* denotes the absence of the primitive state. The probabilities of the system starting in states $\neg$*Leak* and *Leak* are p_1 and p_2 respectively[1], where $0 \leq p_1, p_2 \leq 1$ and $p_1 + p_2 = 1$. The probability of the system to stay burning within one time unit is p_{11}. The probability of flame failure within one time unit is p_{12}. So $0 \leq p_{11}, p_{12} \leq 1$ and $p_{11} + p_{12} = 1$. The probability of the detector to detect the leakage (thereby causing re-ignition the flame) within one time unit is p_{21}. The probability with which the detector fails to detect the leakage within one time unit is p_{22}, where $0 \leq p_{21}, p_{22} \leq 1$ and $p_{21} + p_{22} = 1$. Here we assume that the transition probabilities are independent of the transition history. This is the main feature of a Markov chain.

3.2 Unreliable Detector and Ignition

An implementation with more imperfect components is modelled by a larger graph. The model of a Gas Burner with an unreliable flame detector and an unreliable ignition can be illustrated in Figure 2. This graph uses two primitive states *gas* (the gas and ignition is switched on), and *flame* (the flame is on) to model the system:

- It starts in the idle state, i.e. both the gas and the flame are off,

$$p_1 = 1 \quad \text{and} \quad p_2 = p_3 = p_4 = 0$$

- It idles with probability p_{11} for one time unit;
- The ignition succeeds with probability p_{12} within one time unit;
- The ignition fails with probability p_{13} within one time unit;
- The system stays burning with probability p_{22} within one time unit;
- The system finishes service with probability p_{21} within one time unit;
- The system detects and stops a failure (by returning to the idle state) with probability p_{31} within one time unit;

[1] We usually assume that the Gas Burner starts from the state $\neg$*Leak*, i.e. $p_1 = 1$ and $p_2 = 0$.

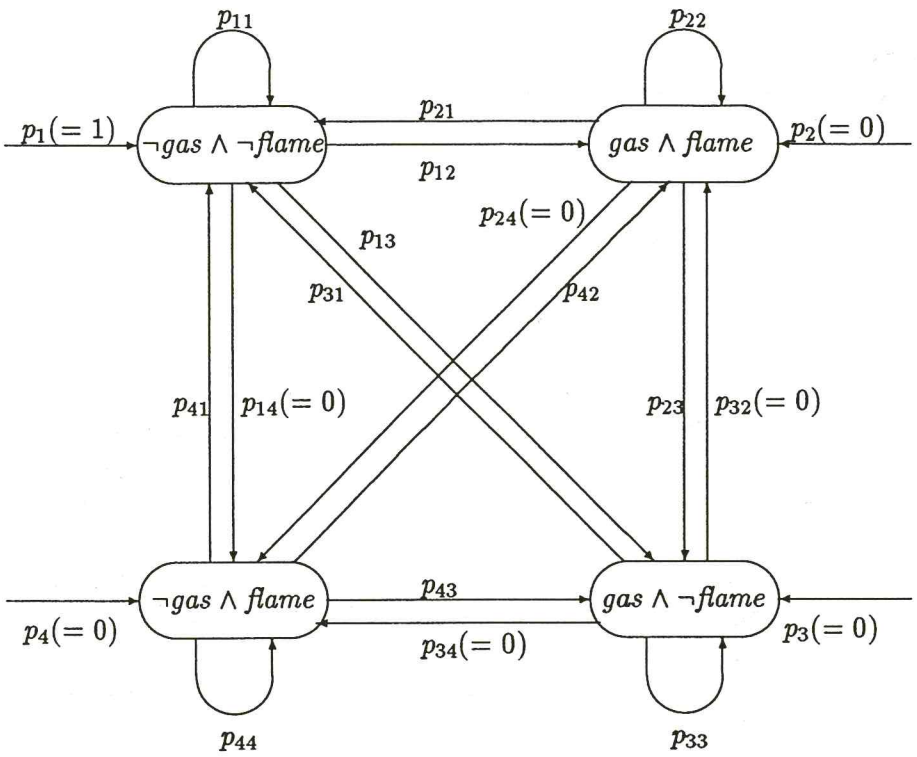

Figure 2: A Gas Burner With Unreliable Detector and Ignition

- The flame fails with probability p_{23} within one time unit;
- Detector or recovery fails with probability p_{33} within one time unit.

We have $0 \leq p_{ij} \leq 1$ and

$$p_{i1} + p_{i2} + p_{i3} + p_{i4} = 1 \qquad (i = 1, 2, 3, 4)$$

Leak is now the composite state

$$Leak \triangleq gas \wedge \neg flame$$

We have used *complete graphs* in the figures to illustrate the models. For simplicity, we can eliminate *unreachable* states such as the one labelled by $\neg gas \wedge flame$ in Figure 2, to which the initial probability and the probabilities of transitions to it are zero, and eliminate all the impossible transitions which depart from a unreachable state or have probabilities zero (see Section 6.2).

3.3 Probabilistic Automaton

We end this section with a general definition of a probabilistic automaton (PA).

Definition 5 *A PA is a tuple* $G = (A, \tau_0, \tau)$ *where*

- A *is a finite non-empty set of primitive states.*

  A basic conjunction[2] *of* A *defines a state of the system, which determines the presence and absence of the primitive states. The set of all basic conjunctions is denoted by* V*, ranged over by* v*,* v'*,* v_i*, etc.*

- $\tau_0 : V \to [0,1]$ *is a function called the initial probability mass function, such that*

$$\sum_{v \in V} \tau_0(v) = 1$$

  $\tau_0(v)$ *defines the probability of the system starting from state* v*.*

- $\tau : V \times V \to [0,1]$ *is a function called the single-step probabilistic transition function such that for every* $v \in V$

$$\sum_{v' \in V} \tau(v, v') = 1$$

For example, in Section 3.1, $A = \{Leak\}$, $V = \{Leak, \neg Leak\}$. The initial probability mass function is $\tau_0(\neg Leak) = 1$, $\tau_0(Leak) = 0$. And the probabilistic transition function is as follows.

$$\tau(\neg Leak, \neg Leak) = p_{11} \qquad \tau(\neg Leak, Leak) = p_{12}$$

$$\tau(Leak, \neg Leak) = p_{21} \qquad \tau(Leak, Leak) = p_{22}$$

4 Satisfaction Probability

Given a $G = (A, \tau_0, \tau)$ where V is the set of all basic conjunctions of A.

4.1 Probabilistic Behaviour

An infinite sequence of states in V,

$$\sigma : v_1, \ldots, v_n, \ldots,$$

defines a possible *behaviour* of G. That is, start from state v_1 and transit from state v_{i-1} to v_i within one time unit. Let X denote the set of all behaviours of G, ranged over by σ, γ, etc. We use $\sigma^{[n]}$ to denote the prefix of length n of σ, and $X(\sigma^{[n]})$ to denote the set of behaviours with common prefix $\sigma^{[n]}$. Note that $X = X(\sigma^{[0]})$ for any σ.

[2] A conjunction of A is *basic*, if every primitive state in A or its negation , but not both, appears in the conjunction.

τ_0 determines the probability of the set of behaviours starting from an initial state, and τ determines the probability of a transition from one state to another. Therefore, τ_0 and τ together determine the probability of the set $X(\sigma^{[n]})$ of behaviours with the common prefix $\sigma^{[n]}$ of length n. We denote its probability by $\mu(X(\sigma^{[n]}))$. Thus

$$\mu(X(\sigma^{[n]})) \triangleq \tau_0(\sigma(1)) * \sum_{i=1}^{n-1} \tau(\sigma(i), \sigma(i+1))$$

where $\mu(X(\sigma^{[0]}) \triangleq 1$ and $\mu(X(\sigma^{[1]})) \triangleq \tau_0(\sigma(1))$

For example, let $\sigma^{[1]} = Leak$ in Section 3.1. $X(\sigma^{[1]})$ is then the behaviours starting from state *Leak*. But $\tau_0(Leak) = 0$, i.e. the system cannot start from state *Leak*. Thus, $\mu(X(\sigma^{[1]}) = 0$. Similarly, let

$$\sigma^{[5]} : \neg Leak, \neg Leak, Leak, Leak, \neg Leak$$

be another prefix. The probability of behaviours with $\sigma^{[5]}$ as common prefix is

$$\mu(X(\sigma^{[5]})) = p_1 * p_{11} * p_{12} * p_{22} * p_{21}$$

We now define a probability space $\langle X, \mathcal{A}, \mu \rangle$, where $\mathcal{A}$ is the sigma-algebra[3] generated by the sets $X(\sigma^{[n]})$. The definition of μ on the generators $X(\sigma^{[n]})$ given above uniquely determines the *probabilistic measure* which is the extension of μ (also denoted μ) to all sets in the sigma-algebra $\mathcal{A}$ [Fel66]. This definition of the probability space makes the probabilistic automaton a discrete Markov chain [Fel66].

4.2 Satisfaction

A behaviour σ of G determines presence and absence of the primitive states at each time point, and thus defines an interpretation I_σ of duration formulas with A as the primitive states. That is,

$$I_\sigma(P)(j) \triangleq \begin{cases} 1 & \text{if } \sigma(j) \Rightarrow P \\ 0 & \text{if } \sigma(j) \Rightarrow \neg P \end{cases}$$

where $j \in N$.

This gives the following definition of satisfaction of a duration formula D (with A as primitive states) by a behaviour σ of G with a time interval $[t_1, t_2]$,

$$\sigma, [t_1, t_2] \models D \quad \text{iff} \quad I_\sigma, [t_1, t_2] \models D$$

For example, let

$$\sigma^{[5]} \triangleq \neg Leak, \neg Leak, Leak, Leak, \neg Leak$$

Then for every behaviour $\gamma \in X(\sigma^{[5]})$, we have

$$\gamma, [0,5] \models \int 1 = 5, \quad \gamma, [0,5] \models \int \neg Leak = 3, \quad \gamma, [0,5] \models \Box(\lceil Leak \rceil \Rightarrow \int 1 \leq 2)$$

$$\gamma, [0,5] \not\models \int 1 \leq 3, \quad \gamma, [0,5] \not\models \Box(\lceil Leak \rceil \Rightarrow \int 1 \leq 1)$$

[3] That is, $\mathcal{A}$ is a family of subsets of X such that: (i) If $X_1 \in \mathcal{A}$ so is its complement $X - X_1$. (ii) If $\{X_n\}$ is any countable collection of sets in $\mathcal{A}$, then also their union $\bigcup X_n$ and their intersection $\bigcap X_n$ belong to $\mathcal{A}$ [Fel66].

4.3 Satisfaction Probability

The probability of a PA satisfying a requirement (a duration formula) within a certain operating time (starting from time zero) is defined as the probability of the set of behaviours of the system which satisfy the requirement within this interval. Let D be a duration formula, and $[0,t]$ be a prefix interval of N. Then $\mu(D)[0,t]$, denoting the *satisfaction probability* of D by G within the time interval $[0,t]$, is defined

$$\mu(D)[0,t] \triangleq \mu(\{\sigma \in X \mid \sigma, [0,t] \models D\})$$

Note that whether a behaviour σ satisfies D within $[0,t]$ only depends upon its prefix $\sigma^{[t]}$ of length t. Therefore, the set on the right hand side of the above definition is the union of a finite number of generators, say $X(\sigma_1^{[t]}), \ldots, X(\sigma_m^{[t]})$, of the sigma-algebra $\mathcal{A}$ in the probability space $\langle X, \mathcal{A}, \mu \rangle$, or it is an empty set. That is, it is a member of $\mathcal{A}$.

For example, for the PA defined in Section 3.1, let $D \triangleq \Box(\lceil Leak \rceil \Rightarrow \int 1 \leq 1)$. Then within $[0,2]$, the behaviours satisfying D are the union of the sets

$$X(\neg Leak, \neg Leak) \quad , X(\neg Leak, Leak) \quad , X(Leak, \neg Leak)$$

Thus

$$\mu(D)[0,2] = p_1 * p_{11} + p_1 * p_{12} + p_2 * p_{21} = p_{11} + p_{12} = 1$$

5 PDC: Probabilistic Duration Calculus

This section establishes a calculus about the satisfaction probability $\mu(D)$, and the length T of a prefix interval over N.

The *probabilistic duration logic* is based on a first order modal logic [HC84] and real arithmetic with additional interval functions T and $\mu(D)$s. Let *Int* be the set of the finite prefix intervals over N. The function T belongs to $Int \rightarrow N$ and assigns each prefix interval $[0,t]$ with its length t, while $\mu(D)$ belongs to $Int \rightarrow [0,1]$ which assigns each prefix interval $[0,t]$ with the satisfaction probability of duration formula D within $[0,t]$.

Therefore, in this logic a *primitive term* is T, $\mu(D)$, or a variable x ranging in the real numbers. A *term* is a primitive term, or an expression of terms built using the usual operators on real numbers, such as addition $+$ and multiplication $*$, with their standard meanings.

A *primitive formula* is an expression built from terms using the relational operators, such as equal $=$ and less than $<$ with their standard meanings.

A *formula* is a primitive formula or an expression built from formulas using the first order logic operators, a modal operator $\Diamond_p$ and the quantifiers over variables.

The modal operator $\Diamond_p$ is interpreted as: $\Diamond_p F$ holds for an interval $[0,t]$ iff there is a prefix $[0,t_1]$, $t_1 \leq t$, such that F holds for $[0,t_1]$. We assume the standard interpretations for the rest of the logic operators and quantifiers.

In this logic, we can write down and reason about probabilistic formulas such as

$$\mu(\neg \textbf{Req-1}) \leq \mu(\neg \textbf{Des-1}) + \mu(\neg \textbf{Des-2})$$

which asserts that the probability of violating the requirement will not be greater than the sum of the probabilities of violating the design decisions. This formula tells the designer that there is a trade off between the design decisions with respect to probabilities. It also allows the designer to consider the reliability of each one separately.

Satisfaction probabilities can also be calculated with this logic by reasoning about formulas of the form

$$T = t \Rightarrow \mu(D) = p$$

As an extension, PDC will include all axioms and rules from the modal logic and real arithmetic. We present in what follows the additional ones for T and $\mu(D)$.

The duration formula *true* defines the set X of all behaviours of G for any interval.

AR 1 *For the duration formula true*

$$\mu(true) = 1$$

For any given interval, the sets of behaviours defined by D and $\neg D$ form a partition of all the behaviours X. So the sum of their probabilities is 1.

AR 2 *For an arbitrary duration formula D*

$$\mu(D) + \mu(\neg D) = 1$$

The following axiom formalizes the additivity rule in probability theory.

AR 3 *For arbitrary duration formulas D_1 and D_2*

$$\mu(D_1 \vee D_2) + \mu(D_1 \wedge D_2) = \mu(D_1) + \mu(D_2)$$

The satisfaction probability is monotonic in the sense that

AR 4 *If $D_1 \Rightarrow D_2$ holds in the duration calculus, then $\mu(D_1) \leq \mu(D_2)$ holds in PDC.*

The above four axioms and rules follow directly from probability theory. The following theorem can easily be proven from them.

Theorem 1 *For arbitrary duration formulas D, D_1, D_2 and D_3*

1. $\mu(false) = 0$
2. $0 \leq \mu(D) \leq 1$
3. *If $D_1 \Leftrightarrow D_2$ in the duration calculus, then $\mu(D_1) = \mu(D_2)$*
4. *If $D_1 \wedge D_2 \Rightarrow D_3$ in the duration calculus, then*

$$(\mu(D_1) = 1) \Rightarrow (\mu(D_2) \leq \mu(D_3))$$

No behaviour with a prefix of length t satisfies $\int 1 \neq t$.

AR 5 $T = t \Rightarrow (\mu(\int 1 \neq t) = 0)$

The validity of a *global formula* (no modal terms T or $\mu(D)$ in it) does not depend upon the time interval. More generally we have

AR 6 *For a PDC formula F such that t is not free in F or F is a global formula, if $(T = t) \Rightarrow F$ holds, then F holds.*

Within $[0, t]$, D and $(D \wedge \int 1 = t)$; *true* are satisfied by the same behaviours.

AR 7 *For an arbitrary duration formula D*

$$T = t \Rightarrow (\mu(D) = s) \quad \textit{iff} \quad T \geq t \Rightarrow (\mu(D \wedge (\int 1 = t);\ \textit{true}) = s)$$

$\Diamond_p F$ means a prefix segement satisfying F. So we have the rule:

AR 8 *For an arbitrary PDC formula F without u as a free variable,*

$$\exists u \leq t : T = u \Rightarrow F \quad \textit{iff} \quad T \geq t \Rightarrow \Diamond_p F$$

We now can prove the following theorem.

Theorem 2 *For arbitrary duration formulas D, D_1 and D_2*

1. $T = t \Rightarrow (\mu(D) = \mu(D \wedge \int 1 = t))$
2. $T = t \Rightarrow (\mu(\int 1 = t) = 1)$
3. *If* $\mu(D_1) = 0$, *then* $\mu(D_1;\ D_2) = 0$

The axioms and rules described so far are independent of the Markov properties of the PA defined by the probability space $\langle X, \mathcal{A}, \mu \rangle$. We consider, in this paper, only those PAs which are Markov chains. The two following axioms formalize the Markov properties for a PA $G = (A, \tau_0, \tau)$.

AR 9 *For an arbitrary state $v \in V$,*

$$T = 1 \Rightarrow (\mu(\lceil v \rceil\ 1) = \tau_0(v))$$

Here we have used the convention $\lceil v \rceil^1 \triangleq \lceil v \rceil \wedge (\int 1 = 1)$.

This axiom formalizes the initial probability mass function τ_0. The probabilistic transition function τ is formalized as follows.

AR 10 *For an arbitrary duration formula D and states $v_i, v_j \in V$,*

$$\mu(D;\ \lceil v_i \rceil^1;\ \lceil v_j \rceil^1) = \tau(v_i, v_j) * \mu(D;\ \lceil v_i \rceil^1;\ \int 1 = 1)$$

This provides a way for calculating the probability of behaviours by chopping of unit intervals.

As mentioned before, for a given interval $[0, t]$, the set of behaviours satisfying a duration formula D can be partitioned into the union of a finite number of generators of the sigama-algebra $\mathcal{A}$. Each generator $X(\sigma_i^{[t]})$ can be defined by a duration formula

$$\lceil v_{i1} \rceil^1;\ \ldots;\ \lceil v_{it} \rceil^1$$

where $v_{ij} = \sigma_i^{[t]}(j)$. We need a structure induction rule to formalize this fact.

Let $R(X)$ be a PDC formula, where X is a variable ranging over duration formulas. R is said to be *disjunction closed*, if $R(X \vee Y)$ is provable from $R(X)$ and $R(Y)$ assuming that $X \wedge Y \Leftrightarrow \mathit{false}$.

AR 11 *Let $R(X)$ be disjunction closed.*

1. *If $R(\lceil\ \rceil)$ and $R(\mathit{false})$ hold, and $R(X;\ \lceil v \rceil^1)$ is provable from $R(X)$ for any $v \in V$, then $R(D)$ holds for any duration formula D.*
2. *If $R(\lceil\ \rceil)$ and $R(\mathit{false})$ hold, and $R(\lceil v \rceil^1;\ X)$ is provable from $R(X)$ for any $v \in V$, then $R(D)$ holds for any duration formula D.*

Using this rule, we can prove the following Markov property.

Theorem 3 *For arbitrary duration formulas D, D_1 and D_2 and any state $v \in V$, if*

$$\mu(D_1;\ \lceil v \rceil^1) = \mu(D_2;\ \lceil v \rceil^1)$$

then

$$\mu(D_1;\ \lceil v \rceil^1;\ D \wedge (\textstyle\int 1 = r)) = \mu(D_2;\ \lceil v \rceil^1;\ D \wedge (\textstyle\int 1 = r))$$

This says that the satisfaction probability of a formula after a state $v \in V$ only depends on the state v, but not on what has happened before v.

Theorem 4 *For any duration formulas D_1 and D_2 and nodes $v_i, v_j, v_k \in V$,*

$$\tau(v_i, v_j) * \tau(v_j, v_k) * \mu(D_1 \wedge (\textstyle\int 1 = r);\ \lceil v_i \rceil^1;\ \lceil v_k \rceil^1;\ D_2;\ \int 1 = 1)$$

$$= \quad \tau(v_i, v_k) * \mu(D_1 \wedge (\textstyle\int 1 = r);\ \lceil v_i \rceil^1;\ \lceil v_j \rceil^1;\ \lceil v_k \rceil^1;\ D_2)$$

This theorem gives a way to calculate the probability from the middle of a behaviour.

6 Examples

6.1 A Gas Burner

Consider the Gas Burner illustrated in Section 3.1. We show how to estimate the probability of the requirement. We assume one time unit to be one second and take the result in [ZHR92],

$$(\textbf{Des-1} \wedge \textbf{Des-2}) \Rightarrow \textbf{Req-1} \quad (\text{i.e. } \neg\textbf{Req-1} \Rightarrow (\neg\textbf{Des-1} \vee \neg\textbf{Des-2}))$$

From **AR** 3 and **AR** 4, we then have

$$\mu(\neg\textbf{Req-1}) \leq \mu(\neg\textbf{Des-1} \vee \neg\textbf{Des-2}) \leq \mu(\neg\textbf{Des-1}) + \mu(\neg\textbf{Des-2})$$

$$\mu(\neg\textbf{Des-1}) = \mu(\mathit{true};\ (\lceil \mathit{Leak} \rceil \wedge (\textstyle\int 1 > 1));\ \mathit{true})$$

$$\mu(\neg\textbf{Des-2}) = \mu(\mathit{true};\ ((\lceil \mathit{Leak} \rceil;\ \lceil \neg \mathit{Leak} \rceil;\ \lceil \mathit{Leak} \rceil) \wedge (\textstyle\int 1 < 32));\ \mathit{true})$$

In what follows, we present a recursive calculation of $\mu(\neg$**Des-1**$)$. ¿From the duration calculus,

$$\neg\textbf{Des-1} \wedge \textstyle\int 1 \leq 1 \Leftrightarrow \mathit{false}$$

Therefore, by Theorems 2.1, 1.3 and 1.1,

$$T \leq 1 \Rightarrow \mu(\neg\textbf{Des-1}) = 0$$

Also, $(\neg$**Des-1** $\wedge \int 1 = 2) \Leftrightarrow \lceil \mathit{Leak} \rceil^1;\ \lceil \mathit{Leak} \rceil^1$; but $\tau_0(\mathit{Leak}) = 0$ thus

$T = 2 \Rightarrow \mu(\neg$**Des-1**$) = 0$, by Theorems 2.1, 1.3 and **AR 10** and **AR 9**.

Des-1 is violated for the first $t+1$ time units, $t > 1$, if and only if **Des-1** has been violated for the first t time units already, or **Des-1** holds for the first t time units but it is violated one time unit later. These two cases are mutually exclusive. The probability of the first case can be recursively calculated. The second case is formulated by duration formula

$$(\textbf{Des-1};\ \textstyle\int 1 = 1) \wedge \neg\textbf{Des-1} \wedge (\int 1 = t + 1)$$

that is equivalent to

$$(\textbf{Des-1};\ \lceil \neg \mathit{Leak} \rceil^1;\ \lceil \mathit{Leak} \rceil^1;\ \lceil \mathit{Leak} \rceil^1) \wedge (\textstyle\int 1 = t + 1)$$

which describes how the last unit's *Leak* violates **Des-1**.

All the arguments above can be formalized in the duration calculus, but are not completely presented here. By Theorem 1 and **AR 3** and **AR 10**, the probability of the second case can be calculated recursively by the probability of $(\textbf{Des-1};\ \lceil \neg \mathit{Leak} \rceil^1)$

$$\mu((\textbf{Des-1};\ \lceil \neg \mathit{Leak} \rceil^1;\ \lceil \mathit{Leak} \rceil^1;\ \lceil \mathit{Leak} \rceil^1) \wedge (\textstyle\int 1 = t + 1))$$

$$= \quad p_{12} * p_{22} * \mu((\textbf{Des-1};\ \lceil \neg \mathit{Leak} \rceil^1;\ \textstyle\int 1 = 2) \wedge (\int 1 = t + 1))$$

Now we introduce two functions $\mathcal{P}(t)$ and $\mathcal{Q}(t)$ defined respectively by

$$(T = t \wedge t > 2) \Rightarrow (\mu(\neg\textbf{Des-1}) = \mathcal{P}(t))$$

$$(T = t \wedge t > 2) \Rightarrow (\mu(\textbf{Des-1};\ \lceil \neg \mathit{Leak} \rceil) = \mathcal{Q}(t))$$

As discussed above, with the calculus we can prove

$$\begin{cases} \mathcal{P}(t+1) = \mathcal{P}(t) + p_{12} * p_{22} * \mathcal{Q}(t-1) \\ \mathcal{Q}(t+1) = p_{11} * \mathcal{Q}(t) + p_{12} * p_{21} * \mathcal{Q}(t-1) \end{cases}$$

where $\mathcal{P}(t)$ gives us the concrete value for $\mu(\neg\textbf{Des-1})$ for a given value of t.

The calculation of $\mu(\neg\textbf{Des-2})$ is given recursively as follows. From **AR** 2,

$$\mu(\neg\textbf{Des-2}) = 1 - \mu(\textbf{Des-2})$$

And in the duration calculus,

$$(\textbf{Des-2} \wedge \textstyle\int 1 > 0) \Leftrightarrow (\textbf{Des-2} \wedge (\mathit{true};\ \lceil \mathit{Leak} \rceil^1)) \vee (\textbf{Des-2} \wedge (\mathit{true};\ \lceil \neg \mathit{Leak} \rceil^1))$$

So by **AR 3** and Theorem 1.1, we have

$$\begin{aligned} T > 0 \Rightarrow (\mu(\textbf{Des-2}) &= \mu(\textbf{Des-2} \wedge (\mathit{true};\ \lceil \mathit{Leak} \rceil^1)) \\ &\quad + \mu(\textbf{Des-2} \wedge (\mathit{true};\ \lceil \neg \mathit{Leak} \rceil^1))) \end{aligned}$$

Let $\mathcal{U}(t)$ and $\mathcal{V}(t)$ be the functions such that

$$T = t \Rightarrow \mu(\textbf{Des-2} \wedge (\mathit{true};\ \lceil \mathit{Leak} \rceil^1)) = \mathcal{U}(t)$$

$$T = t \Rightarrow \mu(\textbf{Des-2} \wedge (\mathit{true};\ \lceil \neg \mathit{Leak} \rceil^1)) = \mathcal{V}(t)$$

We can derive the following recursive equations for $\mathcal{U}(t)$ and $\mathcal{V}(t)$ in the calculus.

$$\begin{cases} \mathcal{U}(t+1) = p_{22} * \mathcal{U}(t) + \begin{cases} p_{11}^{28} * p_{12} * \mathcal{V}(t-29) & \text{if } t > 29 \\ p_{11}^{t-1} * p_{12} & \text{if } 1 \le t \le 29 \\ 0 & \text{if } t < 1 \end{cases} \\ \mathcal{V}(t+1) = p_{21} * \mathcal{U}(t) + p_{11} * \mathcal{V}(t) \end{cases}$$

Using the above mutually recursive equations, we can calculate $\mu(\textbf{Des-2})$ and thus $\mu(\neg\textbf{Des-2})$.

As an illustration we have calculated that, to ensure that the requirement **Req-1** with a probability no less than 0.999 within one day (24 hours); the implementation components must be so chosen that the transition probabilities p_1, p_2, p_{11}, p_{12}, p_{21} and p_{22} guarantee

$$\mu(\neg\textbf{Des-1}) + \mu(\neg\textbf{Des-2}) \le 0.001$$

within one day.

6.2 A Protocol Over an Unreliable Communication Medium

Consider a *medium* through which a *sender* process sends messages to a *receiver* process. To describe the behaviour of the protocol, we introduce the following states.

- s, b, m and r represent that the sender, buffer, medium and receiver are active respectively. e represents an error state of the medium.
- The protocol starts from state s, i.e. the sender is active to send a message.
- The sent message is written into a buffer within one time unit.
- The medium receives the message from the buffer within one time unit.

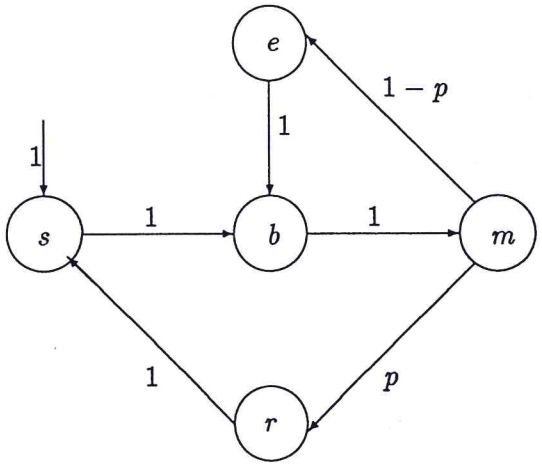

Figure 3: A Protocol Over an Unreliable Medium

- Within one time unit, the message sent by the medium may be received by the receiver with probability p and the protocol enters state r, or the message is lost with probability $1-p$ and the protocol enters error state e.
- If the message is lost, within one time unit, the medium re-reads the message from the buffer.
- If the message is received by the receiver, within one time unit, the receiver acknowledges the sender and the sender is ready to send another message.

The protocol is illustrated in Figure 3. The primitive states which make up the composite states are not elaborated and the unreachable states and transitions with probability 0 are eliminated.

The first kind of properties we are interested in are the so called *soft-deadline* properties [HJ89]. It describes that starting from the state s, within t time units, i.e. within the interval $[0,t]$, the receiver receives at least one message with probability q. This is formalized in terms of PDC as

$$T = t \Rightarrow \mu(\neg(\int r = 0)) = q \quad \text{or equivalently,} \quad T = t \Rightarrow \mu(\int r > 0) = q$$

It is not difficult to derive

$$3k < T \leq 3(k+1) \Rightarrow \mu(\neg(\int r = 0)) = 1 - (1-p)^k$$

When $p = 0.9$, i.e. ten percent of the messages are lost, we have $T = 7 \Rightarrow \mu((\int r > 0)) = 0.99$. This gives the same result as presented in [HJ89].

Another kind of properties is to describe the *upper bound* of error occurrences for a given interval $[0,t]$. This property can be specified by the satisfaction probability of $\int e \leq n$, and also reasoned about in the calculus.

Now let us discuss the probability of the *reoccurrence of the error* state. We define the following formulas for shorthands.

$$D_1 \triangleq (\int 1 = k);\ \lceil e \rceil^1;\ \lceil \neg e \rceil \wedge (\int 1 = k_1);\ \lceil e \rceil;\ \mathit{true}$$

$$D_2 \triangleq (\int 1 = k);\ \lceil e \rceil;\ true$$

The conditional probability of D_1 under D_2 defines the probability of error reoccurrence in k_1 time units. When $\mu(D_2) \neq 0$, it is equal to $\frac{\mu(D_1 \wedge D_2)}{\mu(D_2)}$, denoted $a(k_1)$.

Using natural induction on k_1, we can derive

$$a(k_1) = \begin{cases} 1-p & \text{if } k_1 = 3 \\ p^{n+1} * (1-p) & \text{if } k_1 4n+6\ (n \geq 0) \\ 0 & \textbf{otherwise} \end{cases}$$

by proving

$$T \geq k + k_1 + 2 \Rightarrow \mu(D_1 \wedge D_2) = a(k_1) * \mu(D_2)$$

7 Conclusion and Discussion

We have presented a probabilistic duration calculus based on discrete Markov processes. The main motivation for the development has been to unify the calculations undertaken in order to verify the correctness as well as reliability of a design. We have illustrated this by analysing the 0.999 probability of failure for a safety requirement **Req-1**

$$T \leq 1day \Rightarrow \mu(\textbf{Req-1}) \geq 0.999$$

for a simple Gas Burner. The analysis uses that a correct design results from two design decisions **Des-1** and **Des-2**. For these formulas the probabilities can be reduced to failure probabilities for components.

The starting point for the development has been the two model approach in [SNH91]; which, however, uses continuous Markov processes. We have chosen discrete Markov processes in order to have a simpler set of rules; but it is planned to investigate a similar extension for continuous time.

The approach in [HJ89], based on CTL in [CES83], can be used to analyse soft-deadline properties. It can also be used to analyse **Des-1**; but we have not succeded in using it to analyse probabilities of **Req-1**, **Des-1**, error occurrences or reoccurrences.

Future work will include investigation of the incorporation of reliability analysis with design for fault-tolerant systems [LJ91, Liu91]. A further essential piece of work is to experiment with schemas for calculating concrete probabilities; because this would allow us to benefit from the existing knowledge about Markov processes in practical applications.

Acknowledgements: We would like to acknowledge helpful discussions with N.H. Hansen, M.R. Hansen and H. Rischel.

References

[CES83] E.M. Clarke, E.A Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specification: A practical approach. In *Proc. 10th ACM Symp. on Principles of Programming Languages*, pages 117–126, 1983.

[Fel66] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 2. John Wiley & Sons, Inc. New York, London, second edition edition, 1966.

[HC84] G. E. Hughes and M. J. Cresswell. *A Companion to Modal Logic.* Methuen and Co., London, 1984.

[HJ89] H. Hansson and B. Jonsson. A framework for reasoning about time and reliability. In *Proc. 10th IEEE Real-Time System Symposium, S:a Monica, Ca.*, 1989.

[HRR91] K.M. Hansen, A.P. Ravn, and H. Rischel. Specifying and verifying requirements of real-time systems. In *ACM SIGSOFT '91 Conference on Software for Critical Systems*, December 1991.

[HZ92] M. R. Hansen and C.C. Zhou. Semantics and completeness of duration calculus. Technical report, ProCoS, 1992. To appear in *Real-Time: Theorem in Practice*, J.W. de Bakker, W.-P. de Rover and G. Rozenberg (Eds.) in the LNCS-series.

[Liu91] Z. Liu. *Fault-Tolerant Programming by Transformations.* PhD thesis, Department of Computer Science, University of Warwick, 1991.

[LJ91] Z. Liu and M. Joseph. Transformation of programs for fault-tolerance. *Formal Aspects of Computing*, To appear, 1991.

[LS89] K. Larsen and A. Skou. Bisimulation through probabilistic testing. In *Proc. 16th ACM Symposium on Principles of Programming Languages*, 1989.

[MSS82] P. M. Melliar-Smith and R. L. Schwartz. Formal specification and mechanical verification of SIFT: A fault tolerant flight control system. *IEEE Trans. on Computers*, 31(7), 1982.

[RR91] A.P. Ravn and H. Rischel. Requirements capture for embedded real-time systems. In *IMACS-IFAC Symposium MCTS, Lille, France*, pages vol. 2, pp. 147–152, 1991.

[SNH91] E.V. Sørensen, J. Nordahl, and N.H. Hansen. From CSP models to Markov models: a case study. Technical Report (Submitted for publication in IEEE Transactions on Software Engineering), Institute of Computer Science, Technical University of Denmark, 1991.

[SRRZ92] J.U. Skakkebæk, A.P. Ravn, H. Rischel, and Chaochen Zhou. Specification of embedded, real-time systems. In *EuroMicro Workshop on Formal Methods for Real-Time Systems (submitted)*, 1992.

[ZHR92] C.C. Zhou, C.A.R. Hoare, and A.P. Ravn. A calculus of durations. *Information Processing Letters*, 40(5):269–276, 1992.